# On Growth of Parallelism within Routers and Its Impact on Packet Reordering[1]

A. A. Bare[1], A. P. Jayasumana, and N. M. Piratla[2]
Department of Electrical & Computer Engineering, Colorado State University, Fort Collins, CO 80523, USA
[1]Agilent Technologies, Inc., 900 S. Taft Av., Loveland CO 80537, USA
[2]Deutsche Telekom Laboratories, Ernst-Reuter-Platz 7, D-10587 Berlin, Germany

*Abstract*— **The network link speeds increase at a higher rate compared to processing speeds. This coupled with the increase in size of router tables demand higher levels of parallelism within router hardware. However, such parallelism introduces unintended consequences that potentially may negate some of the performance gains provided by the improved technology. The growth trends of computing speeds, link speeds, and routing table sizes are used to evaluate one such consequence, packet reordering within routers. Results presented show the trends related to the degree of hardware parallelism and packet reordering**

## I. INTRODUCTION

As the speed of physical links and networks increase beyond gigabit per second, and the end-to-end latency to packet transmission time ratio increases by orders of magnitude, certain phenomena that were insignificant and safely ignored assume substantial importance. In fact some of these second order effects, unless countered, can negate to a significant degree the gains provided by faster physical links and routing/switching hardware, and will have an adverse impact on the end-to-end performance seen by the applications. These unavoidable phenomena include, among others, delay jitter and packet reordering. Jitter has received attention only with respect to real-time applications such as VoIP, and effects of reordering were safely ignored.

According to Moore's law, the CPU computing speed approximately doubles every 18 months [1, 2], while recent trends indicate that network link speed approximately doubles every nine months [3, 4]. Thus, the network link speeds increase at a faster rate than the computing speed. The Internet itself is growing in size, resulting in the increase of routing tables sizes in backbone routers [5]. Consequently, the amount of processing to be performed by the routers will increase at a faster rate than the rate of increase in the computing power. Routers will rely on architectures that use an increasing number of processors working in parallel to counter the additional computation requirements. However, processing packets from the same stream in parallel processors deteriorates the problem of reordering. The two obvious solutions, avoiding parallel processing of packets of the stream by sending all of them (or at least those that have the potential for reordering) to the same processor, and buffering packets and forcing them to leave in-order, pose challenges that require additional resources.

In this paper, we analyze the impact of the increasing link-speed vs. CPU performance gap and the growth of router table sizes on parallelism required within routers, and the resulting effect on reordering. By scaling the CPU speed, link speed, routing table size and the number of flows, we show that the parallelism in routers has to increase significantly, an unintended and inevitable result of which is reordering. While the present generation of high-performance routers compensate for internal reordering within routers using input tracking and output buffering techniques, such techniques do not scale well with the increasing performance gap, and result in higher router latency. This paper, while not presenting solutions to this dilemma, attempts to quantify the parallelism and make a case for dealing with such secondary effects proactively in hardware architectures and protocols as speeds continue to evolve.

Section II reviews the router functionality and architecture, and addresses the impacts of packet reordering. The rates of change of CPU speeds, link speeds and routing table sizes are related to parallelism within routers in Section III. Section IV presents simulation-based trends for packet reordering for different levels of parallelism. Conclusions are in Section V.

## II. BACKGROUND ON ROUTERS AND REORDERING

### *A. Routers*

A router performs two basic tasks, route processing and packet forwarding. Routers share information about network conditions, routing information base (RIB), with peers using protocols such as OSPF, RIP, and BGP. Using this information, each router builds and maintains a routing table, which is then used to decide the appropriate outgoing interface for forwarding each incoming packet. The basic functional components of a router are shown in Fig. 1. Switching fabric is the hardware that transfers the packets between the line cards. Line cards are the physical link media whose design depends on the network link technology being

---

[1] This research was supported in part by NSF ITR Grant No.0121546 and by Agilent Technologies.

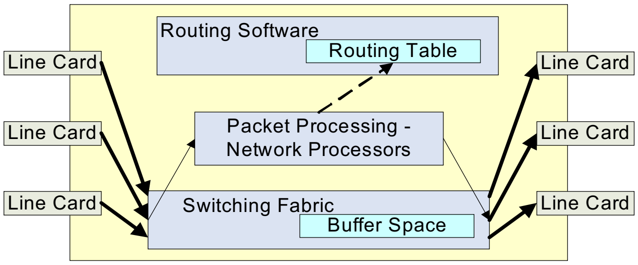

Fig. 1. Essential elements inside a router and the path a packet follows

used, e.g., OC-3. Forwarding of a packet involves a number of subtasks such as extracting the packet header, identifying destination address, finding outgoing interface from the routing table with prefix matching, adding link layer header to the packet for the outgoing link, placing the packets in the corresponding queues and making decisions about dropping packets. Network processor units (NPUs), specialized CPUs such as Intel IXP2400, are used to perform these operations. A high-performance router typically has several network processors, and depending on the architecture it may have a separate NPU per line interface or a pool of NPUs for packets arriving at several interfaces [6,7,8,9]. A single NPU unit may also contain multiple processing engines.

As a packet enters the line input interface of a router, its header is removed and passed through the switch fabric to the packet processor. Using the routing table, this processor determines how to forward the packet and sends back the header after updating it. The line card integrates the new header with the packet and sends the entire packet to the outbound line card [10,11]. All the cells are then sent to the shared memory pool for temporary storage while the IP address is being looked up and the outgoing interface gets ready for sending. This is the “store” part of a “store-and-forward” router. The buffer manager creates a query based upon the information extracted from the packet to determine the outgoing interface of the packet. Query is assigned to an NPU that carries out a longest-prefix match over the routing table. The decision is conveyed to the buffer manager, which sends a notification to the I/O manager of the corresponding outgoing interface. The notification is queued at the outgoing interface. When the notification reaches the head of the queue, the I/O manager reads all the corresponding cells from the shared memory. The checksum and the link layer header are added and the packet is sent over the link. The operations including query formation, routing table lookup, queuing and then sending the packet form the “forward” part of the “store-and-forward” router.

Switch fabric with distributed buffer memory and forwarding [11] is a router architecture in which each line card is equipped with its own forwarding engine to transmit data via a switch fabric to any other line card as needed. This reduces the load on each forwarding engine to that of the corresponding incoming line. However, as the link speeds continue to outstrip processing speeds, parallelism will have to be introduced within the line cards, at which time effects such as reordering introduced by shared memory architectures will appear in routers with this architecture as well.

A complex device such as a backbone router may fail to keep the order of a flow of packets passing through it. Packets in a flow may take different paths. If a packet suffers excessive delay, then it may reach the destination after its successor. This parallelism in processing packets is the primary reason for reordering. Reasons for packet reordering in high-speed routers include following:

- Routers utilize multiple parallel NPUs for meeting performance requirements. A single NPU may even have within itself multiple processing engines, which work on several queries at the same time. Simultaneous processing of packets from the same stream establishes a race condition that may cause the processing of a packet after its successor. Severity of this effect is high when the inter-packet gap among the packets in a stream is low.
- Per-packet load balancing distributes packets towards same destination over multiple outgoing interfaces for even link utilization [7].
- A router may be designed using multiple shared-memory systems or small-sized routers to construct a large-capacity router. Difficulty of coordination among these units causes the order of packets to change [11].
- With head-of-line (HOL) blocking, the outgoing interface of a packet stream may not wait for a blocked packet in the same stream, to sustain interface throughput goals, thereby, causing reordering [11].
- Due to route flapping, the packets in the same stream may be erroneously sent over different paths resulting in out-of-orderliness [12].

In Juniper M160, with four parallel processors, each with a capacity of 2.5 Gbps to serve a single 10Gbps interface, reordering was a concern [13,14]. Corrective action was taken in the design of the subsequent generations of high-end Juniper routers, e.g., T640, to avoid this problem. Other notable instances of reordering include that in BD6808/6816 [14]. The recent high-end routers attempt to reduce or avoid reordering by either a) input buffering, i.e., tracking the packets at the input to identify the individual streams, and forwarding the packets of the same stream to the same queue, thus preventing reordering, or b) output buffering, i.e., buffering packets at the output of the router to ensure that the packets belonging to the same stream are released in the order of their entry into the node [15]. NPUs from vendors such as IBM, Motorola, Vitesse, TI and Intel have built-in hardware to track individual flows [16]. Rearrangement of packets, using input tracking or output buffering, requires identifying each flow, recording the incoming sequence of packets in each flow, and establishing the correct order at the outgoing interface using the recorded sequence. It is also possible to add a time stamp as packets come into the router and buffer and release them without causing reordering. However, as link speed to processing speed increases further, these solutions appear to be non-scalable. Furthermore, buffering packets in the router for taking corrective action adds to the end-to-end packet latency. As the networks move from bandwidth-limited regime to latency-limited regime [17], increasing the latency is not an attractive option. In fact, increase in latency itself may contribute to reduced throughput for many applications, thus negating some of the benefits of increased link speeds.

### B. Impact of Reordering

Recovery from reordering is the responsibility of the end-nodes according to the end-to-end design argument for the Internet as well as the best-effort delivery model. End-point recovery from reordering has worked well in the past with TCP or application-level buffering in case of UDP.

When out-of-order packets are received, TCP perceives it as loss of packets, resulting in deterioration of performance due to following [18]:

- The number of unnecessary retransmissions increases resulting in drop in throughput.
- The congestion window becomes very small due to multiple fast retransmissions, causing problems in raising the window size, resulting in decreased bandwidth utilization.
- Due to multiple retransmissions the round trip time (RTT) is not sampled frequently, thus degrading the estimate of RTT.
- Performance of receiver also suffers, because whenever the receiver receives out-of-order packets, it has to buffer all the out-of-order packets and they need to be sorted as well.
- Detection of loss of packets is delayed because of out-of-order delivery, due to which retransmission request for a lost packet is sent only when TCP times out.
- Due to reverse path reordering, i.e. reordering of acknowledgements, TCP loses its self-clocking property, i.e., property of TCP that it only sends packets when another packet is acknowledged, doesn't remain valid resulting in bursty transmissions and possible congestion.

Packet reordering can severely degrade the end-to-end performance [19]. For certain applications based on UDP, e.g. VoIP, an out-of-order packet arriving after its playback time has elapsed is treated as lost, decreasing the perceived quality of voice on receiver side, but still consuming NIC and processing resources. Corrective action is possible with buffering at the receiver as long as delay is not excessive, but the amount of resources for recovery will increase with the degree and extent of reordering, and with the bit rate of application (e.g., video over IP will require significantly higher bit rates compared to VoIP).

## III. TRENDS IN COMPUTING AND LINK SPEEDS

The processor level parallelism within routers is dictated by the growth rates in link speeds, processing speeds and routing table sizes. In this section, we consider these growth trends and evaluate the parallelism required for future high-performance routers.

### A. Increase in Network Link Speed

Let 'α' be the factor by which the network link speed for a high-end router increases during time period T (months elapsed from some initial $T_0$), i.e., if s is the initial link speed, the link speed will be (s * α) after the period T. As the link speed almost doubles every 9 months,

$$\alpha = 2^{T/9} \qquad (1)$$

### B. Increase in Computing Speed

Let β be the factor of increase in the computing speed during time period T. Applying the Moore's law to the network processors:

$$\beta = 2^{T/18} \qquad (2)$$

### C. Increase in BGP Table Size

Increasing number of subnets in the Internet, and usage of CIDR (Classless Inter-Domain Routing), load balancing, etc., have caused the number of prefixes in the routing table of a generic backbone router to increase almost exponentially between the year 1998 and 2000 [20], and almost quadratically after that [21]. The data obtained from AS1221 [5] is used to estimate the trend corresponding to the size of BGP table. The quadratic equation fitted to data from 2000-2004 is given by:

$$S = 7.804\text{e-}013 * T_u^2 - 0.00076 * T_u + 9.603\text{e+}004 \qquad (3)$$

S is the typical number of BGP entries per router, at the time instance $T_U$, which is in the form of UNIX timestamp. Although long-term observations indicate that a higher-order fit may be better, we use the quadratic relationship. The result would be, if anything, an underestimate of the complexity of routing tables. We use γ to represent the factor by which the size of BGP table increases during the time period T.

### D. Increase in the Number of NPUs

Next we derive a simple approximation for the number of NPUs required (n), after the time duration T, given that the router at the initial time required m NPUs.

Assuming that network usage increases proportionately to network capacity, i.e., the link utilization remains constant, and that the packet lengths remain the same, the number of packets per second arriving over the link also increases by the same factor, α. The amount of work for processing these packets thus increases by the same proportion. The amount of computations required for processing each packet (looking up router entry, etc.) is considered to increase logarithmically with the BGP table size. Thus, the overall amount of computations that the router has to perform increases by a factor

$$\omega = \alpha * \log_2(\gamma) \qquad (4)$$

Considering β as the factor of increase in the computing speed during the time period T, the following relationship can be formulated for m and n:

$$n = \omega * m / \beta \qquad (5)$$

Combining the impacts of scaling on these parameters, we see that the increase in parallelism in routers is inevitable.

### E. Mean Packet Processing Time

The amount of time taken by an NPU for processing a packet is computed using the results of the experiments presented in [22], which were carried out to measure the single-hop delay of a packet through an operational router in a backbone IP network. The router transit time of a packet is observed to be proportional to the length of the packet. This time includes the time spent in the address lookup process, transfer of the packet from the input to the appropriate outgoing interface, and the time spent in the queue at the outgoing interface. The first two of these three operations are mandatory for processing every packet and take a minimum processing time for each packet, as the queuing operation may not be required for each packet. Relationship between the length of a packet and the mean processing time in [22] is:

$$d_p^T(L) = (0.0213*L+25) \qquad (6)$$

where L is the length of the packet in bytes. Thus, $d_p^T(L)$ represents the router transit time (in µs) of a packet of length L bytes, minus the time spent by the packet in the queue at the outgoing interface.

During the time period T, as the computing speed and the processing work increase by factors β, and $\log_2(\gamma)$ respectively, the mean processing time $d_p^T(L)$ taken by a router for a packet of length L bytes is given in µs by:

$$d_p^T(L) = (0.0213*L+25)*\log_2(\gamma)/\beta \qquad (7)$$

## IV. SIMULATION BASED PREDICTIONS

A simulator depicting the functionality of a simple router scaled to handle future generations based on discussions in Section III has been implemented [23]. The goal of the experiments was to study the reordering induced in packet sequences, due to parallel processing of the packets by multiple processors, in future generation routers.

Fig. 2 represents the high level functionality of the simulated router. Multiple packet streams arrive at different line interfaces and are dispatched to an NPU by the dispatcher. Each packet had a sequence number and a stream identifier. One of the randomly selected streams was designated as the stream of interest, which is used for measuring the amount of reordering. Thus, the remaining traffic emulates background traffic, irrespective of the line interface that it entered through. The primary configurable traffic parameters in the simulator were: (a) number of packet streams, (b) nature of the traffic or statistical traffic distribution, (c) packet size distribution, (d) size of packets in the main stream alone, (e) total link bandwidth, and (f) utilization of the link by the aggregate traffic.

Studies have shown that the Internet traffic is self-similar in nature [24]. Thus, the simulator was configured to generate self-similar traffic [25]. Packet size density over the Internet is trimodal with higher frequencies for packet sizes 40-44, 552-576 and 1500 bytes [26,27]. Assuming no jumbo frames, the packet size density follows the trimodal characteristic. The packet size in a given stream was held constant, but packet sizes of different streams were chosen using the tri-modal packet size distribution. The mean number of packets generated per unit time per stream was approximately the same. A line-card input carries a large number of streams, and all the results presented are for a randomly selected stream with packet size 1500B.

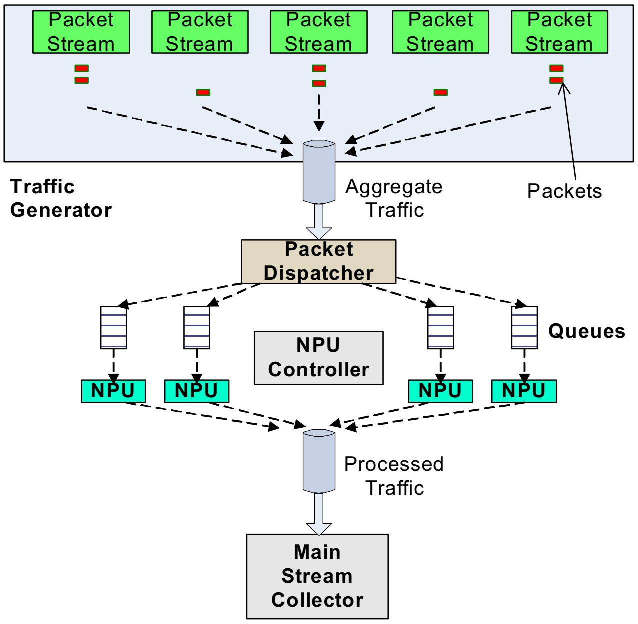

Fig. 2. Functionality of the simulated router based on scaled parameters

The router had a designated number of simulated NPUs, as per Eq. (5), to process the packets arriving at line interfaces. Each NPU was provided with an input queue whose size was limited by the delay-bandwidth product. A packet dispatcher unit distributes the incoming packets to the input queues of the NPUs, using either round-robin scheme or shortest-queue first. As an NPU finished processing a packet, it picks up the next packet in its queue, or waits for a packet to arrive, if the queue is empty. Meanwhile, the processed packet exits the router.

Following the computations of α and β based on T, the values of the factor increase in BGP table size γ, the number of NPUs 'n' using Eq. (5) and the mean processing time $d_p^T(L)$ using Eq. (7) were computed. Table I summarizes the parameter values used for the simulation. The link utilization, unless otherwise stated, was kept at 0.5 during the simulations. In this paper, we present results only for the round-robin scheme for packet dispatcher, which is the more optimistic case as the amount of reordering in this case is lower compared to shortest-queue-first case.

The reordering of the outgoing packets was measured using two metrics, Reorder Density (RD) and Reorder Buffer-occupancy Density (RBD) [28,29,30]. RD is the distribution of the displacements of packets from their original positions, normalized with respect to the number of packets. An early packet corresponds to a negative displacement and a late packet to a positive displacement. RBD is the normalized histogram of the occupancy of a hypothetical buffer that would allow the recovery from out-of-order delivery of packets. If an arriving packet is early, it is added to a hypothetical buffer until it can be released in order. The occupancy of this buffer after each arrival is used as the measure of reordering. A threshold, used to declare a packet as lost, keeps the complexity of computation within bounds. RD and RBD are able to capture reordering more comprehensively compared to existing metrics [31].

| Link change OC-3 to | α | β | γ | ω | n | Mean Packet Processing time |
|---|---|---|---|---|---|---|
| **OC-12** | 4 | 2.00 | 1.60 | 2.71 | 3 | 0.00722 * L + 8.476 |
| **OC-24** | 8 | 2.83 | 1.87 | 7.22 | 5 | 0.00680* L + 7.977 |
| **OC-48** | 16 | 4.00 | 2.16 | 17.77 | 9 | 0.00592 * L + 6.944 |
| **OC-96** | 32 | 5.66 | 2.46 | 41.56 | 15 | 0.00489 * L + 5.736 |
| **OC-192** | 64 | 8.00 | 2.77 | 94.07 | 24 | 0.00391 * L + 4.593 |
| **OC-384** | 128 | 11.3 | 3.10 | 208.9 | 37 | 0.00307 * L + 3.608 |
| **OC-768** | 256 | 16.0 | 3.44 | 456.3 | 57 | 0.00237 * L + 2.785 |

TABLE I. PARAMETERS USED IN SIMULATIONS FOR DIFFERENT LINK SPEEDS

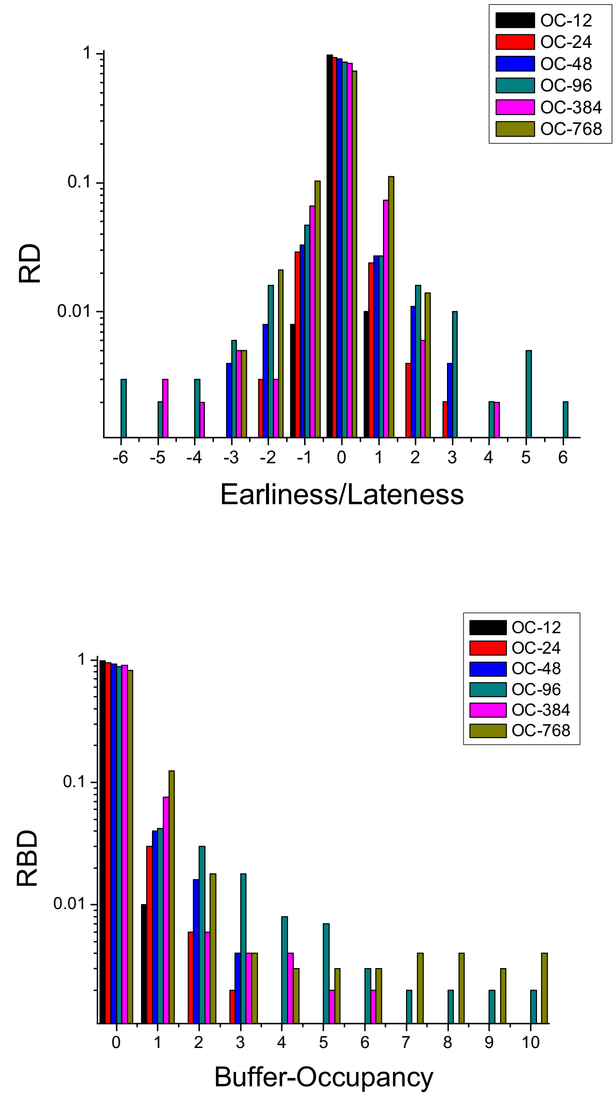

Fig. 3. RD and RBD variation for different incoming link speeds (at 50% utilization)

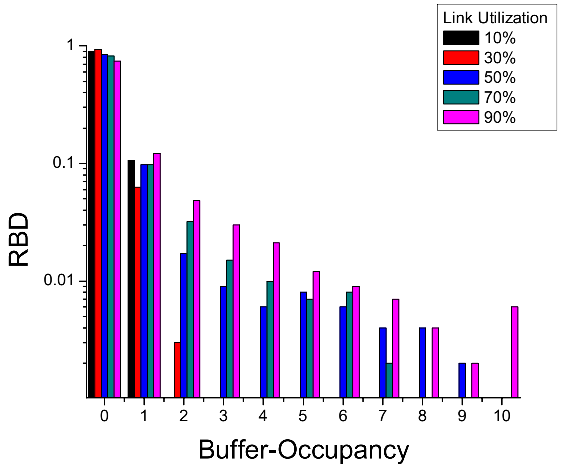

Fig. 4. RBD of reordering through the simulated router for different link utilizations (for OC768, packet size 1500 bytes)

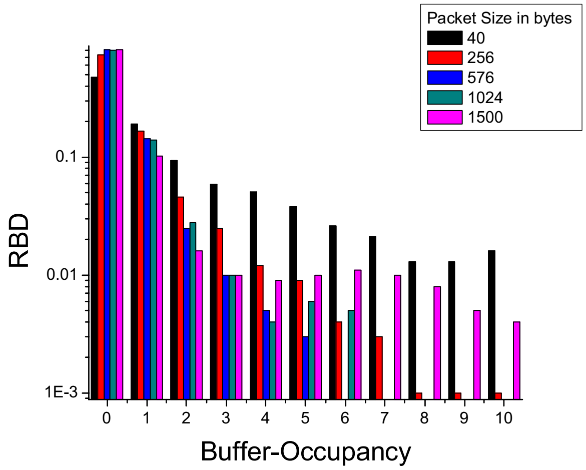

Fig. 5. RBD variation with packet size (with stream of interest occupying 10Mbps on a 50% utilized OC768)

With increasing link speeds and routing table sizes, it is observed that the packet reordering will increase significantly, in the backbone routers. Fig. 3 depicts RBD and RD of the packet sequences in a stream. The RBD indicates that the reorder buffer is occupied by at least one packet 10% of the time for OC-384, and 17% of the time for OC-768. RD indicates that only 84% of the packets arrive at the expected position for OC-384 and only 74% for OC-768. The amount of reordering is much higher than that observed in a typical multi-hop link today despite the fact that reordering due to per-packet scheduling on multiple outgoing links [32] is still not accounted for. Ref [28] shows the RD in a cascade of two subnets to be the convolution of RDs of individual subnets, thus, the overall displacement of a packet from its original position will be significantly higher, when we have multiple routers in the path of a packet stream.

Figures 4 and 5 indicate the variation of reordering, in terms of RBD, of a flow in an OC-768 link with the link utilization and the packet size respectively. Larger packets occupy the link longer, thus helping reduce the amount of reordering. More link utilization aggravates the problem of reordering. From RBD, it is observed that using a buffer for de-ordering, whether attempted at the end-point or within the router with output buffering, will result in significant buffer utilizations as the link speed to processing speed gap increases.

The values in Table 1 should only be considered as indicative of trends in parallelism as opposed to absolute values for each generation of technology. Factors such as possible improvement in router table look up (e.g., hardware implementations), optical switching, etc., could certainly alter the rate of change associated with these trends. Other possible remedies for reducing reordering include increasing the packet length, or even switching bursts of packets instead of individual packets.

## V. CONCLUSIONS

The increasing gap between link and processing speeds and the shrinking packet transmission time with respect to end-to-end latencies will result in second order effects in networks that have been ignored from protocol and performance points of view. We considered packet reordering introduced within routers to show the increasing trends in such second order effects. Countering such effects within routers as well as end-nodes will require increasing resources. Schemes such as load balancing and DiffServ will only increase these effects, which in turn will negatively impact the very same goals these techniques are geared towards, better performance, efficiency and QOS. For example, a load-balancing scheme aimed at reducing overall congestion may result in more reordering, resulting in more retransmissions, thus contributing toward congestion. The need for developing an understanding of these secondary phenomena and their impact on end-to-end performance (as opposed to just the primary effects such as throughput, loss and delay) thus cannot be overemphasized. This topic has received attention only recently, and this is understandable given their negligible impact at sub Gbps rates. There is a lack of understanding of these effects, and no theoretical foundation exists for modeling them let alone predicting them. There is also a need to identify proper metrics for measuring and characterizing these phenomena. Proper understanding of such phenomena can lead to modifications to protocols and architectures that can counter their effects. For example, the acknowledgement transmission policy of TCP may be changed based on measured or estimated values for reordering to overcome deterioration of performance. Tradeoffs involved in recovery from these effects have to be considered as well. Reordering may be dealt at the end nodes with additional buffers, modifications to TCP, etc., at the cost of increase complexity at the end nodes. Alternatively, proactive measures can be taken within routers, such as ensuring in-order release, but these solutions will come at the cost of increased latency. Hardware at the end-node only has to deal with its own flows, while solutions at the router needs to accommodate all the parallel flows passing through each link.

## REFERENCES

[1] G. Moore, “Cramming more components onto integrated circuits,” Electronics, vol. 38, no. 8, Apr. 1965.

[2] Moore’s Law, http://www.intel.com/research/silicon/mooreslaw.htm.

[3] V. Subbiah and P. Pandit, “The unbreakable approach for deploying Oracle9i™ Real Application Clusters,” NetApp Tech Library, TR 3218, Oct. 2002, http://www.netapp.com/library/tr/3218.pdf.

[4] J. Latta, “NGN - The future of networking,” WAVE Report, issue 2054, Nov. 2000, http://www.wave-report.com/other-html-files/NGN1.htm.

[5] BGP Reports, http://bgp.potaroo.net/as1221/bgp-active.html.

[6] M. Kohler, “Network processor overview - Intel, Lucent, Sitera, C-Port,” CMP Media, Inc., 2000, http://www.netrino.com/Articles/NetworkProcessors/.

[7] Cisco, “Load balancing with Cisco express forwarding,” http://www.cisco.com/warp/public/cc/pd/ifaa/pa/much/prodlit/ loadb_an.pdf.

[8] L. Kencl and J.-Y. Le Boudec, “Adaptive load sharing for network processors,” Proc. of INFOCOM 2002, June 2002, pp: 545-554.

[9] Switch Architectures, Light Reading, http://www.lightreading.com /document.asp?site=lightreading&doc_id=25989&page_number=3.

[10] C. Partridge, P. Carvey, et al., "A fifty gigabit per second IP router", IEEE/ACM Transactions on Networking, 6(3), 1998, pp.237-47.

[11] Cisco white paper, “The evolution of high-end router architectures: Basic scalability and performance considerations for evaluating large-scale router designs,” http://www.cisco.com/en/US/products/ hw/routers/ps167/products_ white_paper09186a0080091fdf.shtml.

[12] D. Howe, “Flapping router,” http://burks.brighton.ac.uk/burks/foldoc/18/43.htm.

[13] Internet Core Router Test, http://www.lightreading.com/ document.asp?doc_id=4009&page_number=8.

[14] M. Przybylski, B. Belter, and A. Binczewski, “Shall we worry about packet reordering,” Computational Methods in Science and Technology, 11(2), 141-146, 2005.

[15] H. Liu, “A trace driven study of packet level parallelism,” Proc. of International Conference on Communications (ICC’02), New York, NY, 2002, pp. 2191-2195.

[16] end2end-interest mailing list, “Reordering in Routers,” http://www.postel.org/pipermail/end2end-interest/2003-August/003420.html.

[17] L. Kleinrock, “The latency/bandwidth tradeoff in Gigabit networks,” IEEE Communications, April 1992, 36-40.

[18] M. Laor and L. Gendel, “The effect of packet reordering in a backbone link on application throughput,” IEEE Network, Sept./Oct. 2002, pp. 28-36.

[19] C. Semeria, “Internet Processor II ASIC: Rate limiting and traffic policing features,” White paper, Juniper Networks, 2000. http://www.juniper.net/solutions/literature/white_papers/200005.pdf.

[20] G. Huston, “Analyzing the Internet BGP routing table,” Cisco Systems, http://www.cisco.com/warp/public/759/ipj_4-1/ipj_4-1_bgp.html.

[21] T. Bu, L. Gao and D. Towsley, “On characterizing BGP routing table growth,” Computer Networks, 45(1), 45-54, May 2004.

[22] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, F. Tobagi, and C. Diot, “Analysis of measured single-hop delay from an operational backbone network,” Proc. of IEEE INFOCOM, June 2002, pp: 535-544.

[23] A. A. Bare, “Measurement and analysis of packet reordering,” Masters Thesis, Department of Computer Science, Colorado State University, 2004.

[24] A. Feldmann, A. C. Gilbert, W. Willinger, and T. G. Kurtz, “The changing nature of network traffic: Scaling phenomena,” ACM Computer Communication Review, vol. 28, no. 2, Apr. 1998, pp: 5-29.

[25] G. Kramer, “Generator of self-similar network traffic,” http://wwwcsif.cs.ucdavis.edu/~kramer/code/trf_gen2.html.

[26] K. Claffy, Greg Miller, and Kevin Thompson, “the nature of the beast: recent traffic measurements from an Internet backbone,” CAIDA, http://www.caida.org/outreach/papers/1998/Inet98/Inet98.html.

[27] Caida, Packet Size and Sequencing, http://www.caida.org/analysis/learn/packetsizes/.

[28] N. M. Piratla, A. P. Jayasumana and A. A. Bare, “RD: A formal, comprehensive metric for packet reordering,” Proc. Networking 2005, Lecture Notes in Computer Science 3462, pp. 78-89, May 2005.

[29] A. P. Jayasumana, N. M. Piratla, A. A. Bare, T. Banka, R. Whitner and J. McCollom, “Reorder Density and Reorder Buffer-occupancy Density - Metrics for Packet Reordering Measurements,” IETF draft, Revised April 2007, http://www.cnrl.colostate.edu/Reorder/ draft-jayasumana-reorder-density-07.txt.

[30] N. M. Piratla, A. P. Jayasumana, A. A. Bare and T. Banka, "Reorder Buffer-Occupancy Density and its Application for Evaluation of Packet Reordering," Computer Communications (2007).

[31] N. M. Piratla and A. P. Jayasumana, "Metrics for Packet Reordering - A Comparative Analysis," International Journal of Communication Systems (IJCS), To appear.

[32] N.M. Piratla and A. P. Jayasumana, "Reordering of Packets due to Multipath Forwarding - An Analysis," Proc. IEEE Int. Conf. on Communications (ICC 2006), Istanbul, June 2006.